USN | | | | | | | | | |

Internal Assessment Test 1 – Nov 2024

Sub:	INFORMATION RETRIEVAL					Sub Code:	BAI515B	Branch:	AIML
Date:	06 / 11/2024	Duration:	90 mins	Max Marks:	50	Sem / Sec:	V / A,B,C		OBE

Answer Any of 5 Questions

		MARKS	CO	RBT
1	Explain the Process of Information Retrieval and the components involved in it with a neat architecture.	[10]	CO1	L2
2 (a)	Define the Vector Model and the advantages of the Vector Model?	[05]	CO2	L2
(b)	Can the TF-IDF weight of a term in a document exceed? why?	[05]	CO2	L2
3	Consider the following Documents Query- Obama health Plan D1: Obama rejects allegations about his own bad health D2: The Plan is to visit Obama D3:Obama raises concerns with US health plan reforms Estimate the probability that above Documents are Relevant to the Query.	[10]	CO2	L3
4 (a)	Explain Receiver Operating Characteristics and Benefits of ROC	[05]	CO3	L2
(b)	Consider the Two texts,"Tom and Jerry are friends" and "Jack and Tom are friends". Calculate the Cosine similarity for these two texts?	[05]	CO2	L3
5.a)	Explain the Types of Text Compression Techniques.	[05]	CO3	L2
b)	How Does the Large amount of Information available in Web affect information retrieval system Implementation?	[05]	CO1	L2
6	If an IR System returns 6 relevant Documents and 10 non relevant documents.there are Total of 20 relevant Documents in the collection.calculate the Precision and Recall of the system on this search?	[10]	CO3	L3

CI CCI HOD-AIML

USN | | | | | | | | | | |

Internal Assessment Test 1 – Nov 2024

Sub:	INFORMATION RETRIEVAL				Sub Code:	BAI515B	Branch:	AIML	
Date:	06 / 11/2024	Duration:	90 mins	Max Marks:	50	Sem / Sec:	V / A,B,C		OBE

	Answer Any of 5 Questions	MARKS	CO	RBT
1	Explain the Process of Information Retrieval and the components involved in it with a neat architecture. Definition-4 Drawing- 4 Explanation-2	[10]	CO1	L2
2 (a)	Define the Vector Model and the advantages of the Vector Model? Definition & Examples-3 advantages-2	[05]	CO2	L2
(b)	Can the TF-IDF weight of a term in a document exceed? why? Definition & Examples-3 why TF-IDF-2	[05]	CO2	L2
3	Consider the following Documents Query- Obama health Plan D1: Obama rejects allegations about his own bad health D2: The Plan is to visit Obama D3:Obama raises concerns with US health plan reforms Estimate the probability that above Documents are Relevant to the Query. Stop wor & Stemming-2 M Query Comparison with Documents- 4M Ranking-4M	[10]	CO2	L3
4 (a)	Explain Receiver Operating Characteristics and Benefits of ROC Definition-3M Benefits-2M	[05]	CO3	L2
(b)	Consider the Two texts,"Tom and Jerry are friends" and "Jack and Tom are friends". Calculate the Cosine similarity for these two texts? Definition-3M Steps-2M	[05]	CO2	L3
5.a)	Explain the Types of Text Compression Techniques. Definition-3M Types-2M	[05]	CO3	L2
b)	How Does the Large amount of Information available in Web affect information retrieval system Implementation? Definition-3M Types of Web-2M	[05]	CO1	L2
6	If an IR System returns 6 relevant Documents and 10 non relevant documents.there are Total of 20 relevant Documents in the collection.calculate the Precision and Recall of the system on this search? Definition-5M Steps-2M Precision & recall-3M	[10]	CO3	L3

CI CCI HOD-AIML

USN | | | | | | | | | | |

Internal Assessment Test 1 – Nov 2024

Sub:	INFORMATION RETRIEVAL					Sub Code:	BAI515B	Branch:	AIML
Date:	06 / 11/2024	Duration:	90 mins	Max Marks:	50	Sem / Sec:	V / A, B, C		OBE
Answer Any of 5 Questions							MARKS	CO	RBT

1.	Explain the Process of Information Retrieval and the components involved in it with a neat architecture.	[10]	CO1	L2

A. **Information Retrieval (IR)** can be defined as a software program that deals with the organization, storage, retrieval, and evaluation of information from document repositories, particularly textual information. Information Retrieval is the activity of obtaining material that can usually be documented on an unstructured nature **i.e.** usually text which satisfies an information need from within large collections which is stored on computers. For example, Information Retrieval can be when a user enters a query into the system.

Components of Information Retrieval (IR)

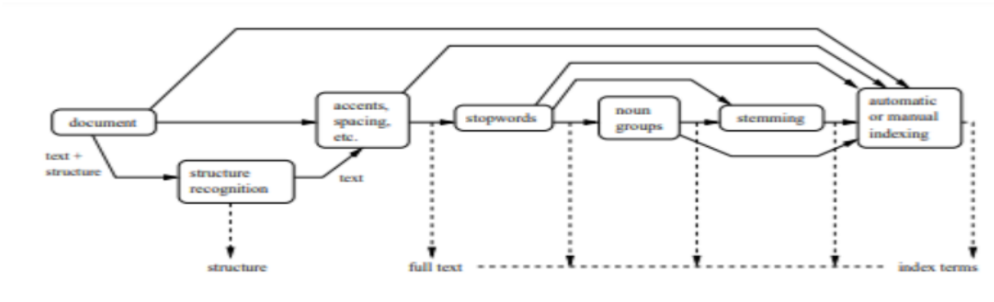


Figure 1.2 Logical view of a document: from full text to a set of index terms.

Document Processing: This is the first step where raw data, like text documents, are prepared for indexing. The goal here is to transform unstructured data into a structured format that is easier to work with.

Document processing includes several sub-steps:

- **Tokenization:** This involves breaking down text into smaller parts, called "tokens." Each token typically represents a word. For example, the sentence "The cat is cute" is split into tokens like "The," "cat," "is," and "cute." Tokenization makes it easier to work with words as individual pieces of data**.**
- **Stemming/Lemmatization:** Words are often reduced to their base or root form. For instance, words like "running," "ran," and "runs" can all be reduced to "run." This helps group similar words and improves the matching between documents and queries.
- **Stop Words Removal:** Common words such as "the," "is," and "and" don't add much meaning to searches, so they're removed. This process helps focus on the essential words that are more likely to determine the document's content.

Indexing: Once the documents are processed, an index is created. Think of indexing like creating a library catalogue where each word points to the documents containing it. An inverted index is commonly used, which maps each word (term) to the documents (IDs) where it appears. This makes searching fast, as we can quickly look up terms and find relevant documents without scanning everything.

Query Processing: When a user submits a query, it's processed to match the indexing structure. This makes the query ready for comparison against the indexed data.

Query processing often includes:

- **Query Tokenization and Expansion:** Similar to document tokenization, the query is split into tokens, and sometimes expanded with synonyms or related words. For example, a search for "car" might include "automobile" or "vehicle" to cover related terms and improve search results.
- **Relevance Feedback:** After an initial set of results, relevance feedback can refine the search. For example, if the user clicks on certain documents, the system may adjust future results to include more similar documents.

Ranking and Retrieval: Algorithms rank documents based on how closely they match the query. Scoring methods like TF-IDF (Term Frequency-Inverse Document Frequency) or BM25 are commonly used. These methods assign scores to documents based on term frequency, rarity, and other factors. Higher scores mean higher relevance, so the system retrieves the most relevant documents first.

Evaluation: This final step assesses the IR system's performance using metrics such as precision (how many retrieved documents are relevant), recall (how many relevant documents were retrieved out of all relevant ones), and the F1-score (a balance between precision and recall). Evaluation helps ensure that the system provides accurate and useful results.

2 (a)	Define the Vector Model and the advantages of Vector Model?	[05]	CO2	L2

In the Vector Space Model (VSM), each document or query is a N-dimensional vector where N is the number of distinct terms over all the documents and queries. The i-th index of a vector contains the score of the i-th term for that vector.

The main score functions are based on: Term-Frequency (tf) and Inverse-Document-Frequency(idf).

Term Frequency (TF)

The **Term Frequency** $\mathrm{tf}_{i,j}$ measures the frequency of the i-th term in the j-th document. It is calculated by dividing the number of occurrences of term iii in document j by the total number of terms in document j:

$$\mathrm{tf}_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

- $n_{i,j}$ is the number of occurrences of term i in document j,
- $\sum_k n_{k,j}$ is the total number of occurrences of all terms in document j.

Inverse Document Frequency (IDF)

The **Inverse Document Frequency** idf_i evaluates the importance of term i across all documents. Rare terms are given higher weights as they are considered more specific to a document. It is computed as:

$$\mathrm{idf}_i = \log \frac{|D|}{|\{d : t_i \in d\}|}$$

where:

- $|D|$ is the total number of documents,
- $|\{d : t_i \in d\}|$ is the number of documents containing term i.

Cosine Similarity

To compute the similarity between two vectors a and b (representing document-query or document-document pairs), we use **Cosine Similarity**. The cosine of the angle between vectors a and b is calculated as:

$$\cos(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\| \|\mathbf{b}\|} = \frac{\sum_{i=1}^{n} \mathbf{a}_i \mathbf{b}_i}{\sqrt{\sum_{i=1}^{n} (\mathbf{a}_i)^2} \sqrt{\sum_{i=1}^{n} (\mathbf{b}_i)^2}}$$

where:

- $\mathbf{a} \cdot \mathbf{b}$ is the dot product of vectors $\mathbf{a}$ and $\mathbf{b}$,
- $\|\mathbf{a}\|$ and $\|\mathbf{b}\|$ are the magnitudes (norms) of vectors $\mathbf{a}$ and $\mathbf{b}$.

Advantages:

- The retrieval performance is improved by its term-weighting method.
- With the help of its partial matching approach, documents that roughly match the query conditions can be retrieved.
- The documents are sorted using its cosine ranking formula based on how similar they are to the query.

2 (b)	Can the TF-IDF weight of a term in a document exceed? why?	[05]	CO2	L2

1. Term Frequency (TF):

- **Definition**: Term Frequency measures how frequently a term appears in a document. The more frequently a term appears, the more important it is assumed to be for that document.
- **Formula**:

$$TF = \frac{\text{Number of times a term appears in a document}}{\text{Total number of terms in the document}}$$

- **Purpose**: TF helps determine the relevance of a term within a single document. A higher term frequency indicates that the term is more significant in that specific document.

2. Inverse Document Frequency (IDF):

- **Definition**: Inverse Document Frequency measures how unique or rare a term is across all documents in a collection. It helps reduce the weight of common terms (like "the" or "is") that appear in many documents and increases the weight of rare terms.
- **Formula**:

$$IDF = \log\left(\frac{\text{Total number of documents}}{\text{Number of documents containing the term}}\right)$$

- **Purpose**: IDF gives more weight to terms that are specific to fewer documents, which often makes them more relevant to a specific topic.

Yes, the **TF-IDF** weight of a term in a document can, in theory, be **greater than 1**.

1. **High Term Frequency (TF) Contribution**:
   - If a term appears frequently within a document (high term frequency), the TF component (e.g., $\mathrm{tf}_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$) can increase significantly, contributing to a larger overall **TF-IDF** value.
2. **Logarithmic IDF Function**:
   - The **IDF** component is typically a logarithmic function, such as $\mathrm{idf}_i = \log \frac{|D|}{|\{d : t_i \in d\}|}$. If a term is rare across the document corpus, the **IDF** value can be large, especially if the document collection is vast, resulting in a higher TF-IDF score.
3. **No Normalization by Default**:
   - In standard TF-IDF calculation, there's no explicit normalization to constrain weights between 0 and 1, so TF-IDF values can exceed 1 based on the term's relative frequency and rarity.

3	Consider the following Documents Query- Obama health Plan D1: Obama rejects allegations about his own bad health D2: The Plan is to visit Obama D3: Obama raises concerns with US health plan reforms Estimate the probability that above Documents are Relevant to the Query.	[10]	CO2	L3

The keywords from the Query are: “**Obama**”, “**health**”, “**Plan**”.
Here is the **Contingency table** the given documents and query:

Document	Obama	Health	Plan	Total
Doc 1	3/3 = 1	2/3 = 0.67	0/3 = 0	2
Doc 2	3/3 = 1	0/3 = 0	2/3 = 0.67	2
Doc 3	3/3 = 1	2/3 = 0.67	2/3 = 0.67	3
Total	**3**	**2**	**2**	**7**

The probability for the keywords of Query to exist in the documents is given below

$$\text{Probability of Doc 1} = \frac{3 \times 1 + 2 \times 0.67 + 0 \times 2}{7} = 0.62$$

$$\text{Probability of Doc 2} = \frac{3 \times 1 + 0 \times 0.67 + 2 \times 0.67}{7} = 0.62$$

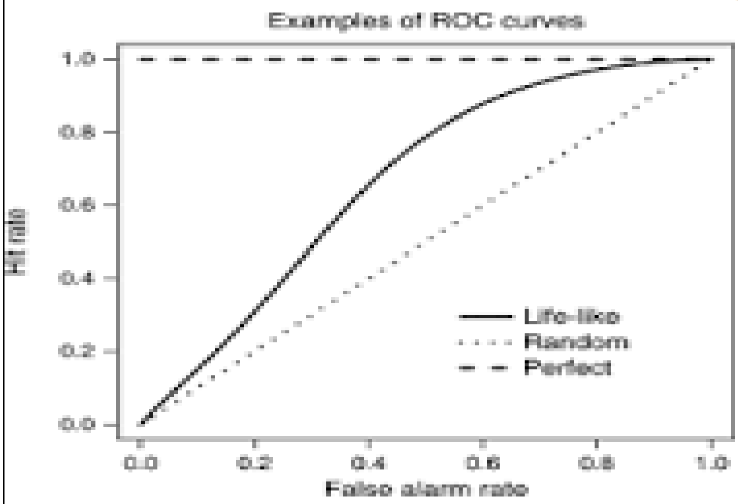
$$\text{Probability of Doc 3} = \frac{3 \times 1 + 2 \times 0.67 + 2 \times 0.67}{7} = 0.9152$$

From the above data it is clear that **Doc3** is having more probability of existence with a probability of **0.9152 (91.52 %)**.
So, the order of relevance is **Doc3 > {Doc1, Doc2} [Since Doc2 and Doc1 are having equal probabilities they are placed in the same set]**

4 (a)	Explain Receiver Operating Characteristics and Benefits of ROC	[05]	CO3	L2

Receiver Operating Characteristics (ROC) is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold varies. It's widely used to evaluate models, particularly in binary classification problems.

1. **True Positive Rate (TPR)** (also known as Sensitivity or Recall):
   - The proportion of actual positives correctly identified by the model.
   - Calculated as $\text{TPR} = \frac{\text{True Positives}}{\text{True Positives + False Negatives}}$.
2. **False Positive Rate (FPR):**
   - The proportion of actual negatives incorrectly identified as positives.
   - Calculated as $\text{FPR} = \frac{\text{False Positives}}{\text{False Positives + True Negatives}}$.
3. **ROC Curve:**
   - The ROC curve is a plot of the TPR against the FPR at various threshold settings. Each point on the curve represents a different threshold for classifying a data point as positive or negative.
   - A model with good performance will have a curve that moves toward the top-left corner, showing a high TPR and a low FPR.
4. **Area Under the Curve (AUC):**
   - The **AUC-ROC** score is the area under the ROC curve. An AUC close to 1 indicates a strong classifier, while an AUC of 0.5 indicates no better than random guessing.



ROC curves are used in many fields, including:

- **Medical diagnostics**: ROC curves are a well-known tool for evaluating the accuracy of diagnostic tests.
- **Epidemiology**: ROC curves can be used in epidemiology.
- **Radiology**: ROC curves can be used in radiology.
- **Bioinformatics**: ROC curves can be used in bioinformatics.
- **Stock market**: ROC curves can be used in the stock market.
- **Fruit tree survival**: ROC curves can be used to predict fruit tree survival.
- **Sports**: ROC curves can be used in sports.

4 (b)	Consider the Two texts, "Tom and Jerry are friends" and "Jack and Tom are friends". Calculate the Cosine similarity for these two texts?	[05]	CO2	L3

Step1: Tokenize the text1

Text1: "Tom and Jerry are friends"

Tokens: ["Tom", "and", "Jerry" "are", "friends"]

Step 2: tokenize the text2

Text2: "Jack and Tom are friends"

Tokens: ["Jack", "and", "Tom", "are", "friends"]

By combining Tokens

Token: [" Tom", "and", "Jerry", "are", "Friends", "Jack"]

Step 3: Prepare term frequency vector

for text1:
V_1: [1, 1, 1, 1, 1, 0]

for text2:
V_2: [1, 1, 0, 1, 1, 1]

Step4:

$$\text{Cosine Similarity} = \frac{A \cdot B}{|A||B|}$$

$$= \frac{(1\times1)+(1\times1)+(1\times0)+(1\times1)+(1\times1)+(0\times1)}{\left(\sqrt{1^2+1^2+1^2+1^2+1^2+0^2}\right)\left(\sqrt{1^2+1^2+0^2+1^2+1^2+1^2}\right)}$$

$= \frac{4}{5} \approx 0.8$

Cosine Similarity = 0.8

5.a)	Explain the Types of Text Compression Techniques.	[05]	CO3	L2

In **Information Retrieval (IR)**, text compression techniques are essential for reducing storage requirements and improving retrieval efficiency. Text compression can be divided into **two main categories**: **lossless** and **lossy** compression. Lossless compression maintains the exact original text, while lossy compression allows some information to be discarded to achieve higher compression. Below are the primary types of text compression techniques used in IR:

1. Statistical Compression

Statistical methods rely on analyzing the frequency of characters or patterns in the text and encoding more frequent items with shorter codes.

- **Huffman Coding**:
  - This is a widely-used lossless technique that builds a binary tree based on character frequency. More frequent characters are given shorter binary codes, and less frequent characters are given longer codes.
- **Arithmetic Coding**:
  - It encodes the entire message as a single number within a range by recursively subdividing intervals based on symbol probabilities. It is highly efficient and can offer better compression rates than Huffman coding.
- **Shannon-Fano Coding**:
  - Similar to Huffman coding, Shannon-Fano coding assigns shorter codes to more frequent symbols. It's not as optimal as Huffman coding but is simpler.

2. Dictionary-Based Compression

These methods replace common phrases or words with shorter codes based on a dictionary.

- **Lempel-Ziv-Welch (LZW)**:
  - LZW builds a dictionary of substrings dynamically as the text is read. It replaces repeated patterns with shorter codes, improving compression for texts with many repeated phrases.
- **Ziv-Lempel (LZ77 and LZ78)**:
  - Both LZ77 and LZ78 use sliding windows to match strings with previously seen sequences. LZ77 references previous occurrences of a string within a defined window, while LZ78 builds a dictionary as it processes text.

3. Transform-Based Compression

These techniques transform the text to improve compression efficiency by rearranging it into a form that's easier to encode.

- **Burrows-Wheeler Transform (BWT)**:
  - BWT rearranges the text so that similar characters are grouped together, which enhances the efficiency of other compression techniques like Run-Length Encoding (RLE) or Huffman Coding. This is the foundation for compression algorithms like **bzip2**.

4. Run-Length Encoding (RLE)

- **Run-Length Encoding**:
  - RLE is a straightforward technique where sequences of repeated characters are stored as a single character followed by a count. For instance, "aaaabbbb" would be encoded as "a4b4". It works well for texts with many repeated characters or patterns.

5. Hybrid Methods

- Some modern compression algorithms combine multiple techniques to maximize compression. For example, the **DEFLATE** algorithm used in ZIP files combines LZ77 (dictionary-based) and Huffman coding (statistical) for enhanced efficiency.

6. Lossy Compression Techniques

- While rarely used for general text due to the need for exact data recovery, lossy techniques can be applied to certain types of IR data, like summarization or topic modeling, where approximate data is acceptable. Techniques like **vector quantization** and **Latent Semantic Analysis (LSA)** are sometimes used in this context.

5 b)	How Does the Large amount of Information available in Web affect information retrieval system Implementation?	[05]	CO1	L2

A **Web Information Retrieval System** is designed to gather, organize, index, and retrieve relevant information from the vast, dynamic content on the internet. Unlike traditional information retrieval systems that work with closed document collections, web IR systems must manage vast data, handle various document formats, and adapt to constantly changing content. Web IR powers search engines like Google, Bing, and others, focusing on speed, relevance, and user experience.

Key Components of a Web Information Retrieval System

1. **Web Crawling**:
   - Web crawlers (or spiders) navigate the internet to discover and retrieve web pages. They follow links to build a comprehensive and up-to-date index of available content, periodically revisiting pages to keep the index current.
2. **Indexing**:
   - After gathering web pages, the system indexes the content, which involves tokenizing text, removing stop words, stemming, and creating an **inverted index**. This index allows for rapid searching by mapping each term to the documents in which it appears.
3. **Document Representation and Metadata Extraction**:
   - Each web page is represented using document vectors (e.g., with TF-IDF or BM25 weighting) that quantify term relevance. Metadata, such as page titles, URLs, and tags, is also extracted to enhance retrieval quality.
4. **Query Processing and Ranking**:
   - User queries are analysed, tokenized, and possibly expanded to match indexed documents effectively. The IR system ranks documents using relevance-based algorithms (e.g., cosine similarity, BM25), link-based ranking (e.g., PageRank), and often considers user intent, context, and personalization factors.
5. **Relevance Feedback and Personalization**:
   - Based on user interactions like clicks and time spent on a page, the system adjusts rankings and tailors' future queries to individual preferences, improving relevance and user satisfaction.
6. **User Interface and Experience**:
   - A user-friendly interface displays search results, query suggestions, filters, and other interactive features. This UI is critical for effective search experience, as it influences how users interact with results and perceive relevance.

The vast amount of information available on the web significantly impacts the **implementation** and **effectiveness** of information retrieval (IR) systems. Here are the key challenges and considerations:

---

1. Scalability and Storage Requirements

- **Challenge**: Web data is constantly growing, so IR systems must handle and store vast amounts of information.
- **Solution**: Efficient indexing, distributed storage systems, and scalable infrastructure (e.g., cloud storage) are required to manage this growth.

2. Speed and Latency in Retrieval

- **Challenge**: Retrieving relevant documents quickly from a large dataset can lead to high latency.
- **Solution**: Optimized indexing (e.g., inverted indexes) and caching are essential to reduce response times and improve user experience.

3. Relevance and Precision of Results

- **Challenge**: Large-scale data leads to diverse content, making it challenging to retrieve only the most relevant results.
- **Solution**: Advanced ranking algorithms, personalized search, and user behavior analysis can help IR systems surface more relevant results.

4. Handling Data Diversity and Quality

- **Challenge**: Web content varies widely in language, structure, quality, and relevance.
- **Solution**: Preprocessing steps like language detection, stopword removal, and content filtering are necessary to handle such diversity.

6	If an IR System returns 6 relevant Documents and 10 non relevant documents.there are Total of 20 relevant Documents in the collection.calculate the Precision and Recall of the system on this search?	[10]	CO3	L3

A

Precision and Recall Calculation for Information Retrieval System

In the context of evaluating the performance of an Information Retrieval (IR) system, **Precision** and **Recall** are two important metrics. Let's calculate these metrics based on the given data:

Given:

- The IR system returns **6 relevant documents**.
- The IR system returns **10 non-relevant documents**.
- The total number of **relevant documents in the collection** is **20**.

The formulae to calculate Precision and Recall are:

1. Precision

Precision is the measure of how many of the documents retrieved by the system are actually relevant.

$$\text{Precision} = \frac{\text{Number of Relevant Documents Retrieved}}{\text{Total Documents Retrieved}}$$

Where:

- The number of relevant documents retrieved = 6.
- The total number of documents retrieved = 16 (6 relevant + 10 non-relevant).

Substituting the values:

$$\text{Precision} = \frac{6}{16} = 0.375$$

Thus, Precision = 0.375 or 37.5%.

This means that 37.5% of the documents retrieved by the IR system are actually relevant.

---

2. Recall

Recall is the measure of how many of the total relevant documents in the collection were successfully retrieved by the IR system.

$$\text{Recall} = \frac{\text{Number of Relevant Documents Retrieved}}{\text{Total Relevant Documents in Collection}}$$

Where:

- The number of relevant documents retrieved = 6.
- The total number of relevant documents in the collection = 20.

Substituting the values:

$$\text{Recall} = \frac{6}{20} = 0.3$$

Thus, Recall = 0.3 or 30%.

CI CCI HOD-AIML